

AN ANALYSIS AND COMPARISON OF PHP7 SUPPORTED FRAMEWORKS

AMITVIKRAM NAWALAGATTI¹ & Dr. PRAKASH R. KOLHE²

¹Lecturer in Computer Science, S K Arts College and H S Kotambri Science Institute,
Hubballi, Karnataka, India

²Research Scholar, Officer Incharge, Agricultural Knowledge Management Unit (AKMU),
Dr Balasaheb Sawant Kokan Krishi Vidyapeeth, Maharashtra, India

ABSTRACT

In the current generation, the rapid development of technologies has been impacting on web development technologies in a positive manner. Developing the framework is one such technology which helps the developers for building the applications. Also selecting the most suitable framework considerably affects the cost and time required for developing the application. This study compares the PHP7 supported frameworks of Code Igniter 3.1.9, Cake PHP 3.6 for the web application of a contact diary. The tests were performed using the Apache J Meter benchmarking tool. Depending on the results of the comparison recommendations are made that help the web developers in selecting the suitable framework for the real world web applications.

KEYWORDS: Web Development, PHP, Framework, Cake PHP & Code Igniter

Received: May 14, 2019; **Accepted:** Jun 26, 2019; **Published:** Jul 20, 2019; **Paper Id.:** IJMCARDEC20194

INTRODUCTION

An effective role is being played by the information technology for the creation of innovation in business. The increased demand for information technology has made the field of web development an important one. An open source language PHP is being widely utilized for developing the websites. PHP help in the development of dynamic websites and has the capability of connecting with the websites. Portability, open source, scalability, performance are some of the features of PHP which make it a great choice for the developers for developing the websites (Supaartagorn, 2011). The use of PHP without its frameworks makes the application maintenance difficult and in turn reducing the performance significantly. In order to resolve such issues, various frameworks have been developed on the internet by web development communities. These frameworks give, distinctive regular functionalities and classes as partners, segments, and module to decrease the time required for the development. Because of these highlights like portability, open source, scalability, performance, these structures are generally utilized for developing websites in PHP, with execution being viewed as the most significant factor.

PHP which was developed in 1995 by RasmusLerdorf is an abbreviation for Hypertext Preprocessor. It is a scripting language which is HTML embedded. RasmusLerdorf utilised Perl for creating PHP since a huge amount of code was required for coding in Perl. PHP is mainly more advantageous when compared to Perl because it was developed for web scripting while Perl was designed for doing many more things. The majority of the syntax for PHP is borrowed from Perl or C with some unique features that are specific to PHP. Also, it is a lot simpler for integrating PHP into HTML rather than Perl. Since its development 19 versions of PHP has been released viz. PHP1, PHP2, PHP3, five versions in PHP4, seven versions in PHP5 and four versions in PHP7. The figure 1 below

illustrates the flow of PHP. In the first step, the client will send the request to the script with the help of a URL. Then the script will process the data and directly send a request to the database. The script will then receive an output from the database for processing of the data. Finally, in the last step (4th) the script will produce the output and the data is sent to the client (Fayyaz, & Munir, 2014).

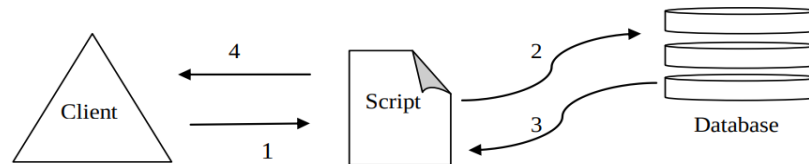


Figure 1: Stages Involved in PHP

The use of frameworks for web programming helps in making the websites and its development much more organised in various ways. The frameworks increase the productivity of programming because the codes that usually require a lot of time to be written can be written in a few minutes using the built-in functions of the framework. The broadly utilized framework has a noteworthy security advantage since its clients become long term analyzers. If a client finds a security issue, he can inform about it through the website of the frameworks since the engineering group would most likely fix it. The majority of the frameworks are available for free or at a very small cost. For all the frameworks, there is a support team and large forums through which the developers can get solutions to their problem easily (Prokofyeva, & Boltunova, 2017). For quite a while PHP programming language was not considered as an adequately genuine language for developing huge Web applications and was considered to be good only for small projects. But in recent times the circumstances has essentially changed. It turned out to be evident that PHP has not just one but many frameworks that extend the capacities of the language. Hence this study evaluates and compares the PHP7 supported frameworks of Code igniter 3.1.9, Cake PHP 3.6 for the web application of a contact diary.

LITERATURE REVIEW

Various studies have been conducted for analyzing and comparing various frameworks of PHP. Mishra, (2014) presented a critical comparison of ASP.NET and PHP for developing a web application of car sales with respect to the features and functional requirements of the application. The results determined that both technologies are similar with respect to programming and are considered suitable for the application. But there are some architectural contrasts including the manner for script compilation and event handling. In relation to popularity, ASP.NET is less popular than PHP. The two technologies have rich APIs for dealing with different database associations. In any case, it very well may be set up that ASP.Net is sensibly perfect with MS SQL Server while PHP with MySQL.

Lancor & Katha, (2013) analysed the PHP frameworks to be utilized for a project that is based on a software engineering course. They conducted an in-depth analysis by the development of a web application with the help of plain PHP as well as other frameworks. The main motivation of this study was the issues that were experienced in developing content-rich programming building course and examines the instructive changes that were made to adjust learning of the student with sound programming designing standards and current programming advancement practices utilized in the industry of computing. The Cake PHP has extensive built-in features but all of them were not applicable for the student projects. Depending on the qualitative and quantitative results the authors concluded that CodeIgniter is the best suited though both CakeP HP and Code Igniter supported most of the functionalities due to the simplicity of the later.

Himawan, (2014) compared the PHP frameworks of Cake PHP and Code Igniter by implementing them in creating the websites for displaying the data from the database. The analysis was done with respect to six factors of each of the frameworks viz. the framework library capacity, implementing through ORM, architectural features, Ajax application, and performance. It was observed that the Code Igniter provided lesser features when compared to Cake PHP, but the authors preferred Code Igniter since it was lighter, could be easily learnt, modified and integrated with the Helper and the Library. Thawari, (2016) computed the performance of CodeIgniter using plain PHP since there was a lack of studies presenting a comparison among plain PHP and Code Igniter. From the comparative analysis, the authors felt that it is better to use Code Igniter PHP rather than plain PHP due to its advantages of instantly fixing the issues of the web applications. Code Igniter PHP also provided interfaces that were user-friendly which helped the developers for creating a secure, flexible and dynamic application in a short duration of time. Most of the studies have been conducted for PHP5 supported frameworks, but none of the studies has been done for PHP 7 supported frameworks hence the main aim of this study is to compare and analyze the PHP7 supported frameworks of Code igniter 3.1.9, CakePHP 3.6 for the web application of a contact diary.

FRAMEWORK ANALYSIS

This study analyses the Code Igniter, Cake PHP frameworks that are fully MCV (Model-View-Controller) compliant, open source and PHP 7 supported. This study developed a contact diary application in Code Igniter and Cake PHP frameworks.

Code Igniter

Code Igniter is basically a toolkit for developing web application utilising PHP. It enables to develop the projects quickly by providing a rich set of libraries for common tasks along with a simple interface and logical structure for accessing the libraries (Betari et al., 2017). The CodeIgniter is based on the MCV development pattern. MCV is a software that is utilised in separating the logic of the application from the presentation. Practically it allows the web [ages to contain less scripting since the presentation and PHP scripting is separate. The main features of CodeIgniter are given below.

- The footprint of the framework is small.
- It is a framework that needs nearly zero configuration and does not need the command line to be used.
- It is compatible with various accounts of standard hosting that run a variety of PHP configurations and versions.
- This framework does not require one to follow the rules of coding strictly.

The architecture of the Code Igniter is illustrated in figure 2 below.

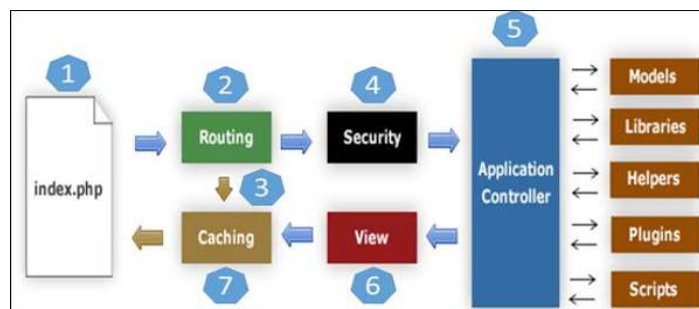


Figure 2: Architecture of Code Igniter

The request cycle of the CodeIgniter starts with the request of the user for a page in the application. Each of the requests will go through the following steps.

- By default, the web server will execute the application/index. php and the base resources that are required for running the CodeIgniter are initialized.
- The HTTP request is checked by the router and it takes a decision on the diversion of the request.
- The cache files, if present, are directly sent to the browser and the response to the request is completed.
- The filtering of the data submitted by the user is done by the security layer.
- The application controller will then load the model, helpers, libraries required for processing the particular request.
- The finalized view is then sent to the web browser for viewing.

The code Igniter skeleton application at the top level comprises of four folders viz. the application, system, tests and User guide_src folders. Figure 3 below shows the folder structure of the Code Igniter.

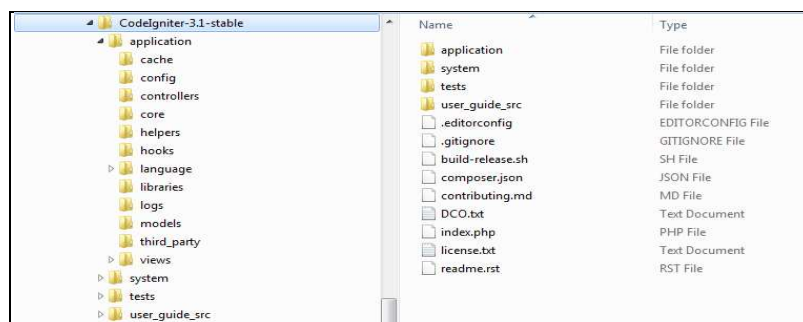


Figure 3: The Folder Structure of the Code Igniter

Cake PHP

CakePHP was developed in by Polish web developer MichałTatarynowicz in 2005 with an aim to make the tasks of web development tasks simpler. This is a language that can be learnt and understood easily. CakePHP is an open source project that is entirely community-driven. Some of the important features of CakePHP are its inbuilt authentication, caching engine, bake console for setting up and running it quickly.

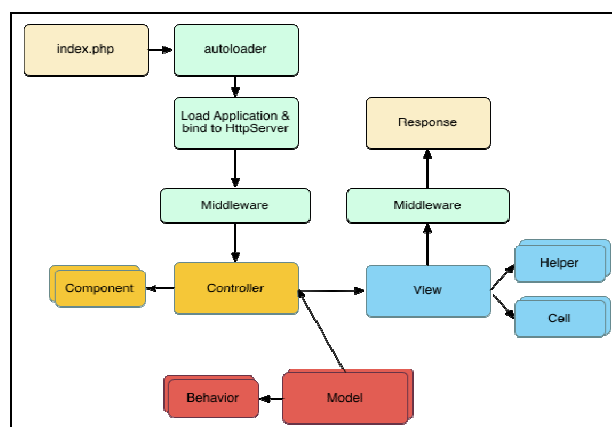


Figure 4: A Cake PHP Request

The structure of the CakePHP request is illustrated in figure 4. The request cycle of the PHP starts with the user requesting for the page in the application. Each of the requests will go through the following steps.

- The web server will first execute the Webroot/index.php.
- The application is then loaded and bound to the HttpServer.
- The middleware of the applications is initialised.
- Through the middleware, the request and the response is dispatched.
- If the request contains information about routing and the middleware does not return any response then the controller and the action is selected.
- Once the controller action is called for, the controller will interact with the necessary components and models.
- The controller then assigns the creation of response to the View for generating the output that is the outcome of the model data.
- View will utilise the Cells and Helpers for generating the headers and body for the response.
- The response is then sent back via the Middleware.
- Finally, the response to the web server is emitted by the HttpServer.

The Cake PHP skeleton application at the top level comprises of nine folders which is shown in Figure 5 below.

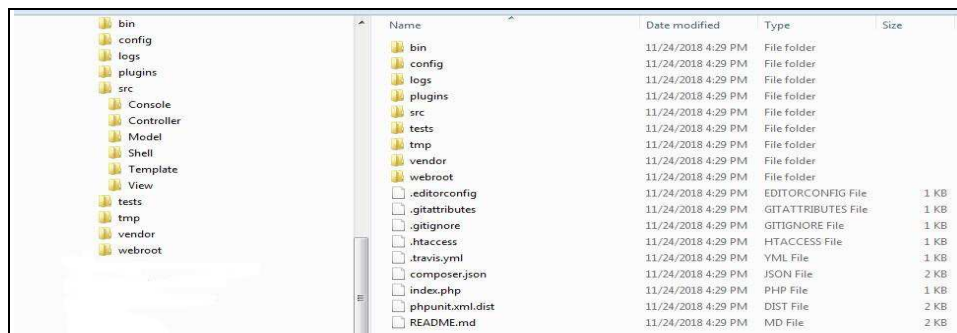


Figure 5: The Folder Structure of the CakePHP

RESULTS AND PERFORMANCE EVALUATION

The web application representing a contact diary wherein each contact has six fields of the first name, last name, email id, mobile number, age and gender was developed in the PHP7 supported frameworks of Codeigniter 3.1.9 and Cake PHP 3.6. The performance of the model was tested while performing the CREATE, READ, UPDATE and DELETE (CRUD) operations. Each of the application has distinct pages for performing the CRUD operations. This simulation conducted eight different operations i. e eight benchmarks for each of the frameworks is done. Each benchmark simulated 100 users requesting the single page 10 times each, hence the total number of requests would be 1000.

Number of Threads (users) = 100

Ramp-up Period (in seconds) = 100 Loop count = 10

Total Requests = 1000 = No of Users * Loop count

The benchmarks are executed utilising the Apache JMeter benchmarking tool and both the applications are deployed on the same. The measurements are taken after 4000 requests so that the JIT's (Just In Time compiler) effects are eliminated. The results obtained for CodeIgniter and CakePHP are given below.

Results of Code Igniter

The home page of CodeIgniter application in figure 6 shows all the contacts, each of them in the form of a card with an image while figure 7 is the table view page where all the contacts are displayed in the form of a table without any images.



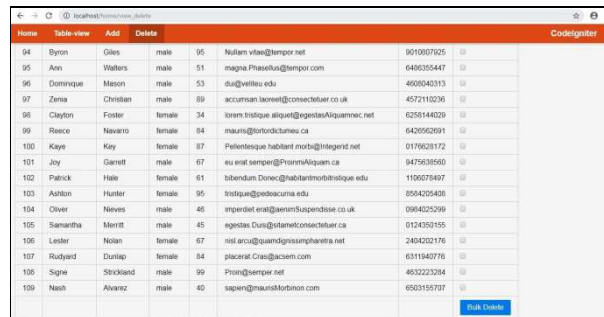
Figure 6: Code Igniter Homepage

ID	First Name	Last Name	Gender	Age	Email	Mobile	Actions
1	ganesh	bansal	male	28	ganesh.bansal@gmail.com	9912345678	View / Edit / Delete
2	rajesh	sharma	male	38	rajesh.sharma@gmail.com	8812345678	View / Edit / Delete
3	sonal	vama	female	58	sonal.vama@gmail.com	8712345678	View / Edit / Delete
4	Dieter	Lang	female	75	nibh.st.amet@molledui.org	7402649004	View / Edit / Delete
5	Blaze	Fields	female	35	diam.nunc@famesac.edu	4767449545	View / Edit / Delete
6	Xenos	Holden	female	62	non@alegiat.co.uk	2944007982	View / Edit / Delete
7	Calvin	Jordan	female	63	lacina.orci@athrus.ca	6830472876	View / Edit / Delete
8	Donovan	Madden	female	86	Sed.congue elit@iacus.org	7971901084	View / Edit / Delete
9	Basil	Velequez	female	59	at.egestas.a@aliquam.net	4423529790	View / Edit / Delete
10	Zeph	Forbes	female	64	metus.eu@semut.ca	3301367776	View / Edit / Delete
11	Barry	Alison	female	79	Fusce.mollis.Duis@Quisquepurusapien.ca	3967851425	View / Edit / Delete
12	Hallee	Young	male	42	amet@acarcu.net	9071396794	View / Edit / Delete
13	Stewart	Potts	female	53	pellentesque.mass@nechaurs.ca	2960590795	View / Edit / Delete

Figure 7: Codeigniter Table-View Page

Figure 8 shows the Add new contact page where the user will have to fill up the contact form and submit it for saving/adding a new contact. Figure 8 is the delete page that is utilised for deleting multiple contacts.

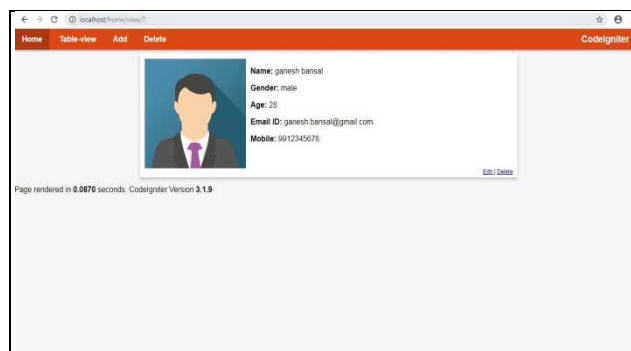
Figure 8: Code Igniter Add a New Contact Page



ID	Name	Email	Delete
94	Byron	Giles	male 95
95	Ann	Walters	male 51
96	Dominique	Marion	male 53
97	Zenia	Christian	male 89
98	Clayton	Foster	female 34
99	Renee	Navarro	female 84
100	Kylee	Key	female 87
101	Joy	Garrett	male 67
102	Patrick	Tate	female 61
103	Ashlon	Hunter	female 95
104	Oliver	Navios	male 46
105	Samantha	Merritt	male 45
106	Lester	Nolan	female 67
107	Rudyard	Dunlap	female 84
108	Signe	Stickland	male 99
109	Nash	Alvarez	male 40

Figure 9: Code Igniter Delete Page

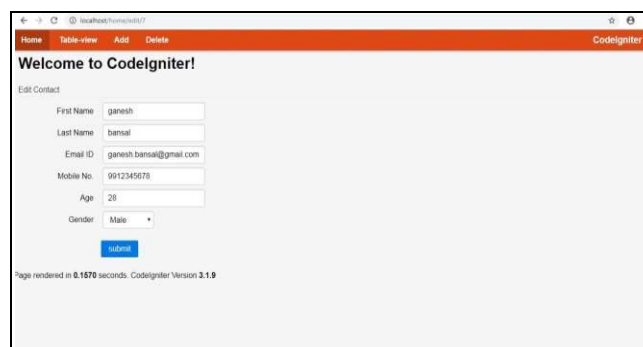
The figures 10 and 11 show the pages for viewing the single contact and for editing the contacts respectively.



Name: ganesh bansal
 Gender: male
 Age: 28
 Email ID: ganesh.bansal@gmail.com
 Mobile: 9912345678

Page rendered in 0.9870 seconds. CodeIgniter Version 3.1.9

Figure 10: Code Igniter View Single Contact Page



Edit Contact
 First Name: ganesh
 Last Name: bansal
 Email ID: ganesh.bansal@gmail.com
 Mobile No: 9912345678
 Age: 28
 Gender: Male

Page rendered in 0.1570 seconds. CodeIgniter Version 3.1.9

Figure 11: Code Igniter Edit Contact Page

Execution Results of Code Igniter

The graphs in figures 12 to 19 represent the results obtained after executing the benchmarks. The table below gives the average, median, deviation and throughput values obtained from the graphs.

Table1

Form	Number of Samples	Average	Median	Deviation	Throughput/Minute
Add form	1000	80	80	5	600.823
Edit form	1000	51	51	5	602.519
Single Contact Card	1000	84	84	5	600.877
Multiple Contact Card	1000	86	82	23	601.504
Bulk delete page	1000	84	83	6	600.781
Insert Contact Request	1000	132	121	34	597.776
Update Contact Request	1000	141	132	49	597.878
Delete Single Contact	1000	153	132	52	597.033



Figure 12: Code Igniter View Add Form



Figure 13: Code Igniter View Edit Form

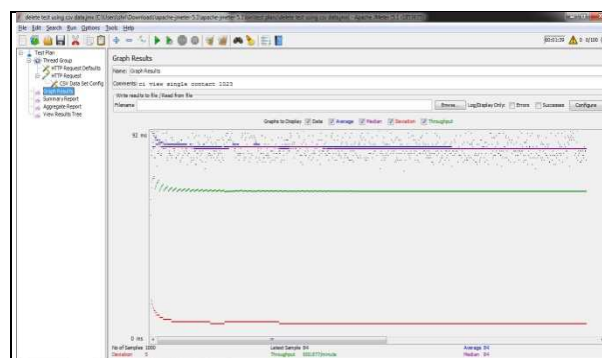


Figure 14: Code Igniter View Single Contact Card

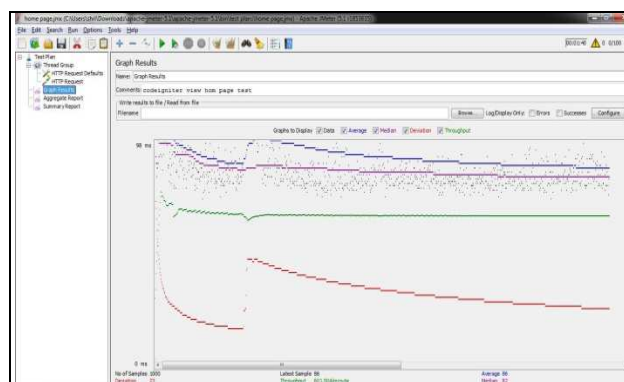


Figure 15: Code Igniter View Multiple Contacts Card

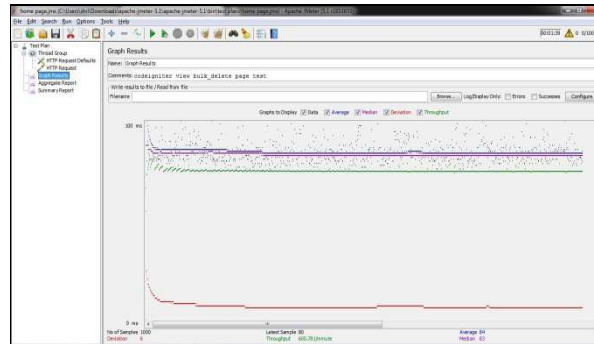


Figure 16: Code Igniter View Bulk Delete Page



Figure 17: Code Igniter Insert Contact Request

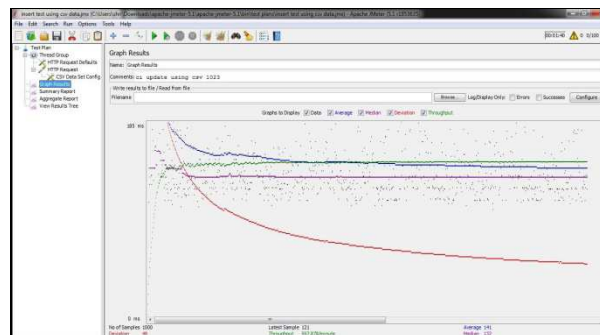


Figure 18: Code Igniter Update Contact Request



Figure 19: Code Igniter Delete Single Contact

Results of Cake PHP

The home page of CakePHP application in figure 20 shows all the contacts, each of them in the form of a card with an image while figure 21 is the table view page where all the contacts are displayed in the form of a table without any images.



Figure 20: CakePHP Homepage

ID	First Name	Last Name	Gender	Age	Email	Mobile	Actions
1	ganesh	bansal	male	28	ganesh.bansal@gmail.com	9912345678	View Edit Delete
2	rajesh	sharma	male	38	rajesh.sharma@gmail.com	8812345678	View Edit Delete
3	sonal	varma	female	58	sonal.varma@gmail.com	8712345678	View Edit Delete
4	Dieter	Lang	female	75	nbn.sil.amet@mollisdui.org	7402649004	View Edit Delete
5	Blaze	Fields	female	35	diam.nunc@famesac.edu	4767448545	View Edit Delete
6	Xenos	Holden	female	62	non@efegiat.co.uk	2944907662	View Edit Delete
7	Kelly	Morrow	male	85	interdum.Curab@convalis.ca	1245214439	View Edit Delete
8	Lucy	Rhodes	female	90	auctor@umarcuVestibulum.net	7062505403	View Edit Delete
9	Calvin	Jordan	female	63	lacinia.orci@atrisus.ca	6630472876	View Edit Delete
10	Donovan	Madden	female	86	Sed.congue.eit@iacus.org	7971901084	View Edit Delete
11	Basil	Velazquez	female	59	at.egestas.a@aliquam.net	4423529790	View Edit Delete
12	Zeph	Forbes	female	64	metus.eu@semut.ca	3301367776	View Edit Delete
13	Barry	Allison	female	79	Fusce.Duis@Quirussapien.ca	3667851425	View Edit Delete

Figure 21: CakePHP Table-View Page

Figure 22 shows the Add new contact page where the user will have to fill up the contact form and submit it for saving/adding a new contact. Figure 23 is the delete page that is utilised for deleting multiple contacts.

Figure 22: CakePHP Add a New Contact Page

ID	First Name	Last Name	Gender	Age	Email	Mobile	Bulk select
1	ganesh	bansal	male	28	ganesh.bansal@gmail.com	9912345678	<input type="checkbox"/>
2	rajesh	sharma	male	38	rajesh.sharma@gmail.com	8812345678	<input type="checkbox"/>
3	sonal	vama	female	58	sonal.varma@gmail.com	8712345678	<input type="checkbox"/>
4	Dieter	Lang	female	75	nbn.sil.amet@mollisdui.org	7402649004	<input type="checkbox"/>
5	Blaze	Fields	female	35	diam.nunc@famesac.edu	4767448545	<input type="checkbox"/>
6	Xenos	Holden	female	62	non@efegiat.co.uk	2944907662	<input type="checkbox"/>
7	Kelly	Morrow	male	85	interdum.Curab@convalis.ca	1245214439	<input type="checkbox"/>
8	Lucy	Rhodes	female	90	auctor@umarcuVestibulum.net	7062505403	<input type="checkbox"/>
9	Calvin	Jordan	female	63	lacinia.orci@atrisus.ca	6630472876	<input type="checkbox"/>
10	Donovan	Madden	female	86	Sed.congue.eit@iacus.org	7971901084	<input type="checkbox"/>
11	Basil	Velazquez	female	59	at.egestas.a@aliquam.net	4423529790	<input type="checkbox"/>
12	Zeph	Forbes	female	64	metus.eu@semut.ca	3301367776	<input type="checkbox"/>
13	Barry	Allison	female	79	Fusce.Duis@Quirussapien.ca	3667851425	<input type="checkbox"/>

Figure 23: Cake PHP Delete Page

The figures 24 and 25 show the pages for viewing the single contact and for editing the contacts respectively.



Figure 24: Cake PHP View Single Contact Page

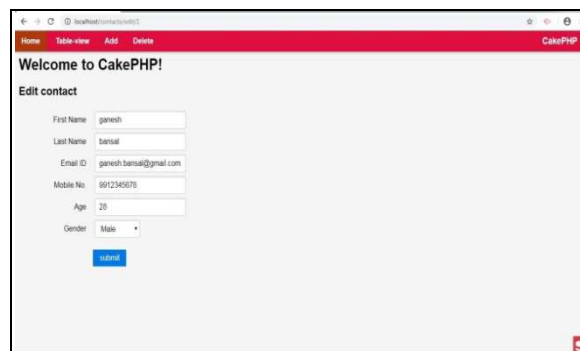


Figure 25: Cake PHP Edit Contact Page

Execution Results of Codeigniter

The graphs in figures 26 to 33 represent the results obtained after executing the benchmarks. The table below gives the average, median, deviation and throughput values obtained from the graphs.

Table 2

Form	Number of Samples	Average	Median	Deviation	Throughput/Minute
Add form	1000	14324	5312	17692	232.359
Edit form	1000	14229	5167	182246	238.57
Single Contact Card	1000	15418	6497	19083	223.033
Multiple Contact Card	1000	16167	6560	19884	224.985
Bulk delete page	1000	17742	7276	20587	204.996
Insert Contact Request	1000	19174	7730	22308	200.496
Update Contact Request	1000	17613	7074	20956	205.935
Delete Single Contact	1000	19836	8441	22791	192.086

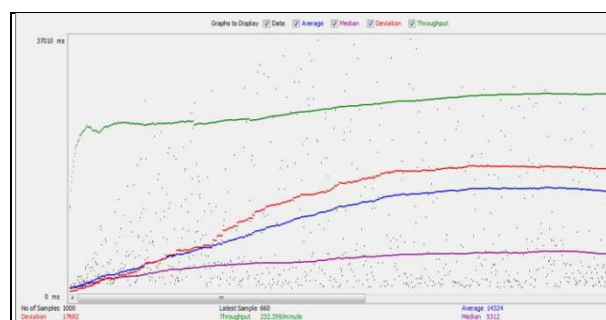


Figure 26: Cake PHP View Add Form Figure

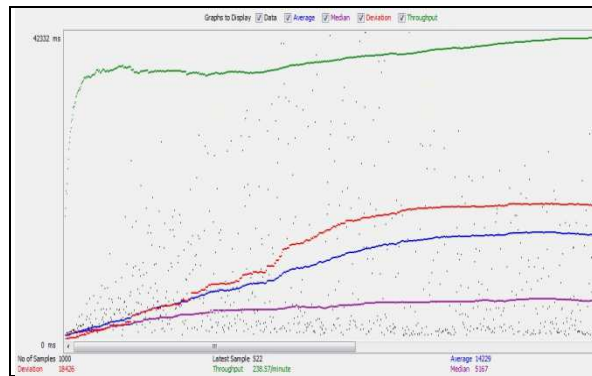


Figure 27: Cake PHP View Edit Form

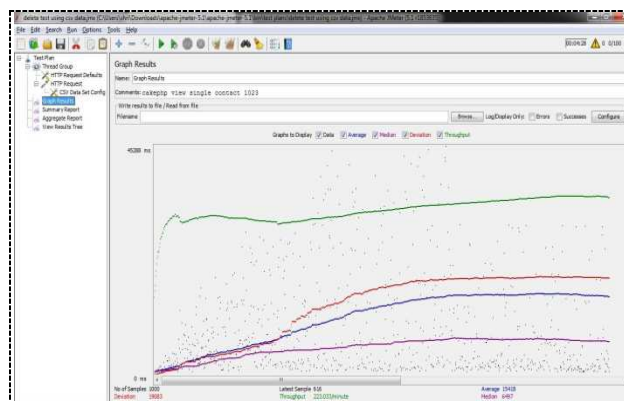


Figure 28: Cake PHP View Single Contact Card

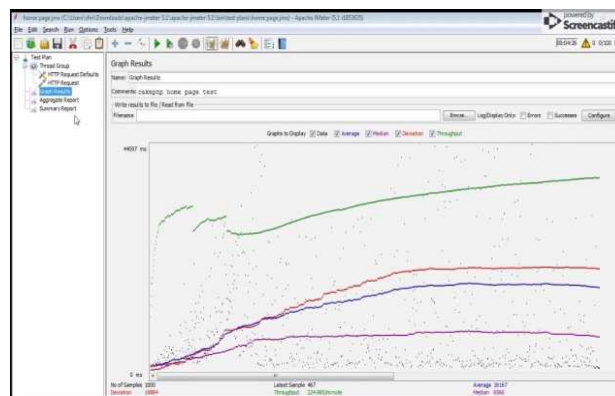


Figure 29: Cake PHP View Multiple Contacts Card

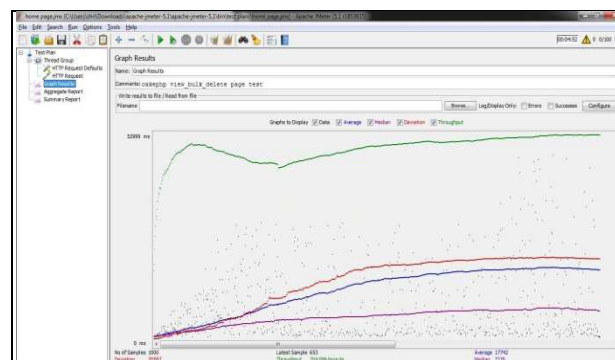


Figure 30: Cake PHP View Bulk Delete Page

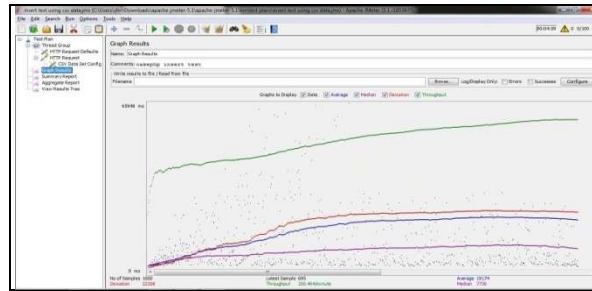


Figure 31: Cake PHP Insert Contact Request

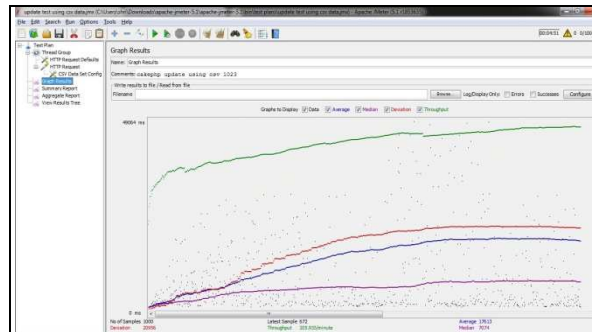


Figure 32: Cake PHP Update Contact Request

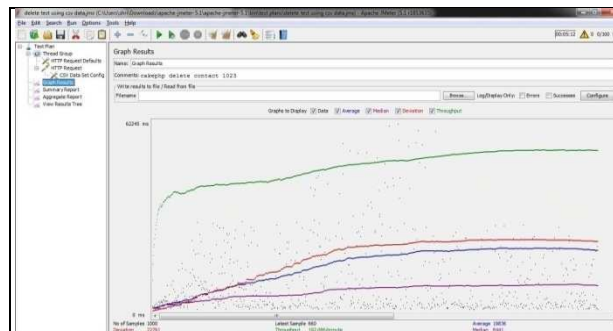


Figure 33: Cake PHP Delete Single Contact

The graphs in figures 34 and 35 provides a comparison of the throughput and average values for both CodeIgniter and CakePHP frameworks. It can be observed that the throughput is higher for CodeIgniter in comparison with CakePHP hence CodeIgniter is more suitable for this application.

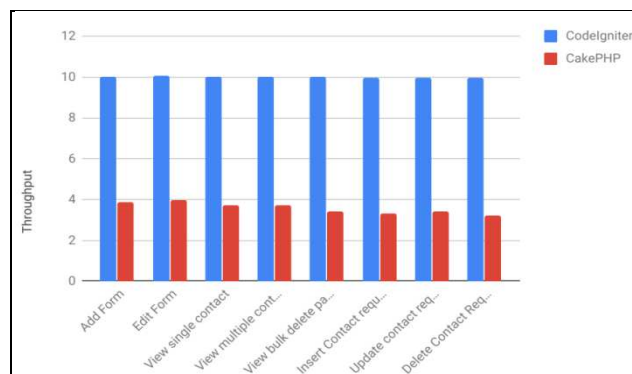


Figure 34: Aggregated Throughput for both Frameworks

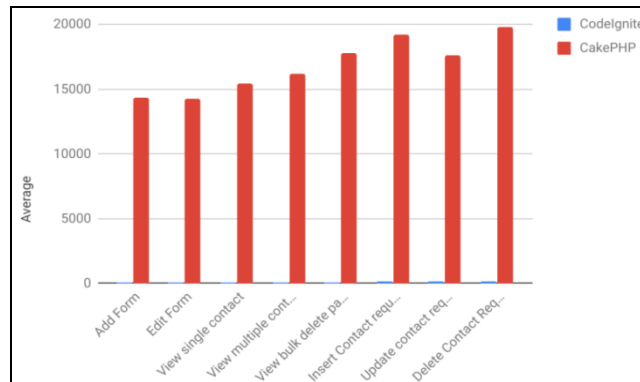


Figure 35: Aggregated Average for both Frameworks

CONCLUSIONS

This study developed a contact diary application in PHP7 supported CodeIgniter and CakePHP frameworks using the Apache JMeter benchmarking tool and compared the two for the suitability of the application with respect to the throughput values. Even though CakePHP provided more features in comparison with CodeIgniter, the latter is preferred since it is easier to learn, modify and integrate. Also the throughput values are higher for CodeIgniter.

REFERENCES

1. Supaartagorn, C. (2011). *PHP Framework for database management based on MVC pattern*. *International Journal of Computer Science & Information Technology (IJCSIT)*, 3(2), 251-258.
2. Fayyaz, A. R., & Munir, M. (2014). *Performance Evaluation of PHP Frameworks (CakePHP and CodeIgniter) in relation to the Object-Relational Mapping, with respect to Load Testing*.
3. Prokofyeva, N., & Boltunova, V. (2017). *Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems*. *Procedia Computer Science*, 104, 51-56.
4. Himawan, A. K. (2014). *Performance Analysis Framework Codeigniter and CakePHP in Website Creation*. *International Journal of Computer Applications*, 975, 8887.
5. Mishra, A. (2014). *Critical Comparison of PHP and ASP.NET for Web Development*. *International Journal of Scientific & Technology Research*, 3(7), 331-333.
6. Brindha, G. (2013). *Performance Analysis of MC-CDMA System Using BPSK Modulation*. *International Journal of Research in Engineering & Technology (IJRET)*, 1(1), 45-52.
7. Thawari, A. S. (2016). *Study Of Codeigniter Technology*. *International Journal of Research in Computer & Information Technology (IJRCIT)*, 1(2), issn: 2455-3743.
8. Lancor, L., & Katha, S. (2013, March). *Analyzing PHP frameworks for use in a project-based software engineering course*. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 519-524). ACM.
9. Betari, O., Erramdani, M., Roubi, S., Arrhioui, K., & Mbarki, S. (2017). *Model transformations in the MOF meta-modelling architecture: from UML to CodeIgniter PHP framework*. In *Europe and MENA Cooperation Advances in Information and Communication Technologies* (pp. 227-234). Springer, Cham